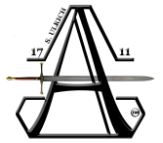


Facer & WatchMaker WatchFaces

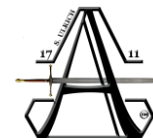
Quick Reference



	Category	Name	Setting	Operator
Watch	Battery	Progress	Fill Ratio	(clamp(#BLN#, 0, 100)/100)
		Progress 0-50	Fill Ratio	(clamp(#BLN#, 0, 50)/50)
		Progress 0-50	Fill Ratio	\$#BLN#>0?(clamp(#BLN#, 0, 50)/50):0\$
		Progress 51-100	Fill Ratio	\$#BLN#>50?(clamp(#BLN#, 50, 100)/50-1):0\$
			Transparency	\$#BLN#>0?100:0\$
			Rotation	((clamp((100-#BLN#),50,100)/100)*360)
			Rotation	((clamp((100-#BLN#),0,50)/100)*360)
			Transparency	\$#BLN#<=20?less:more\$
	Health	Step Count Progress	Fill Ratio	(clamp(#ZSC#, 0, 3000)/3000)
		BPM Progress	Fill Ratio	(clamp(#ZHR#, 0, 200)/200) (Normal Rest HR is normal 60BPM some can be as low as 40, should not be above 100 rest; Max exercise rate 180BPM)
		Miles	Text	((round((#ZSC#*0.0476)))/100) OR ((round((#ZSC#*2.65)/52.8))/100)
		Kilometers	Text	((round((#ZSC#*0.00762)))/10) or ((round((#ZSC#*2.4)/27.7))/100)
		KCAL	Text	(round((#ZSC#/20)))
		(AUTO) Miles/ Kilometers	Text	\$#WM#=F?((round((#ZSC#*0.0476)))/100) m:((round((#ZSC#*0.00762)))/10) km\$
		(AUTO) Miles/ Km Label	Text	\$#WM#=F?((round(#ZSC#))/)Miles:((round(#ZSC#))/)KM\$ For Labels only
		(AUTO) Miles/ Kilometers	Text	(100*(#UNITSYS#==METRIC)) -and- (100*(#UNITSYS#==IMPERIAL)) btw, in miles it will be lower number ((round((#ZSC#*0.00762/1.6)))/10)
	Time	Time Progress Second	Fill Ratio	(clamp(#DWFSS#, 0, 360)/360) (clamp(#DWFSS#, 0, 362)/362) <- To only show 1 sec / this bar covers the second bar with a -1 deg rotation
		Ticks Per Sec		(round(#DWFSS#*X/6)/X*6) where X is the ticks per second. So for 9 ticks per second: (round(#DWFSS#*9/6)/9*6), etc
	Weather	Temp Progress	Fill Ratio	(clamp(#WCT#, 0, 72)/120)
		Forecast Max/Min 1d	Text	#WFBH#º/#WFB#º
		Next Day	Text	\$#DOW#=0?MON:\$#\$DOW#=1?TUE:\$#\$DOW#=2?WED:\$#\$DOW#=3?THU:\$#\$DOW#=4?FRI:\$#\$DOW#=5?SAT:\$#\$DOW#=6?SUN:\$
	Date	Day of Week	Transparency	\$#DOWB#=1?100:0\$ (1-7)
		Split Date	Text	Split the date in to 2 and display each number where ever you want to Get the first number I.E 0-3 (((#Dd#)-(#Dd#%10))/10) to get the second 0-9

Facer & WatchMaker WatchFaces

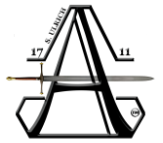
Quick Reference



			(#Dd#%10)
Orbiting	Sun Hour Hand Min Hand Second Hand [Time Based on 24 Hour Rotation – change #DWFHS# for other]	X-position: Y-position:	X-position: $(160 + \cos((\#DWFHS\# + 90) * (\pi / 180))) * 160$ Y-position: $(160 + \sin((\#DWFHS\# + 90) * (\pi / 180))) * 160$ To offset the sun/moon lower during its time increase the + Y point
	Moon (This is 180deg from the other) [Time Based on 24 Hour Rotation]	X-position: Y-position:	X-position: $(160 + \cos((\#DWFHS\# - 90) * (\pi / 180))) * 160$ Y-position: $(160 + \sin((\#DWFHS\# - 90) * (\pi / 180))) * 160$ To offset the sun/moon lower during its time increase the + Y point
Rotation		Rotation:	#DWFSS#
Fade/Color Change	Time Hour	Transparency	Time Hr Night -5 \$#Dk#<5?100:0\$ Time Hr Day +5/-20 \$#Dk#>=5&&#Dk#<=20?100:0\$ Time Hr Night +20 \$#Dk#>20?100:0\$
	Time Sunrise/Sunset	Transparency	Time Hr Night -5 \$#Dk#<#WSUNRISEH24#?100:0\$ Time Hr Day +5/-20 \$#Dk#>=#WSUNRISEH24#&&#Dk#<=#WSUNSETH24#?100:0\$ Time Hr Night +20 \$#Dk#>#WSUNSETH24#?100:0\$
	Day / Night	Transparency	\$#DISDAYTIME#=1?100:0\$ Or \$#DISDAYTIME#==true?100:0\$ (For elements visible at the night use “0” (false) instead of “1” (true) in the condition) and you are done!
		Transparency	\$#Ds#>=1&&#Ds#<12?0:100\$ Or \$#DH#>=0&&#DH#<12?100:0\$ 1-12 opacity is 0 else 100.
	Day / Night	X or Y - position:	night \$#DISDAYTIME#==false?160:999\$ day \$#DISDAYTIME#==true?160:999\$
	Day of week	Transparency	\$#DOW#==1?100:0\$

Facer & WatchMaker WatchFaces

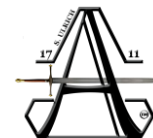
Quick Reference



			0 = Sunday 1 = Monday /etc...
	3k/5k/10k	Transparency	3k $\$ZSC\# < 3000 ? 10000 : 0\$$ 5k $\$ZSC\# \geq 3000 \& \& \#ZSC\# \leq 5000 ? 10000 : 0\$$ 10k $\$ZSC\# > 5000 ? 10000 : 0\$$
	Show During "AM"	Transparency	$\$Da\# == AM ? 100 : 0\$$
	Show During "AM"	Transparency	$\$DH\# < 12 ? AM : PM\$$ or $\$DH\# < 12 ? 100 : 0\$$ for AM and $\$DH\# < 12 ? 0 : 100\$$ for PM
	Heart Rate		HR -45 $\$ZHR\# < 45 ? 200 : 0\$$ HR +45/-100 $\$ZHR\# \geq 45 \& \& \#ZHR\# \leq 100 ? 200 : 0\$$ HR +100/-160 $\$ZHR\# \geq 101 \& \& \#ZHR\# \leq 160 ? 200 : 0\$$ HR +160 $\$ZHR\# > 160 ? 200 : 0\$$
	Next Day Text		Standard Week Day (Sunday (1) to Saturday (7)) (WHITE) $\#DOWB\#$ Week Day + 1 (RED) $((\#DOWB\# + 1) - (7 * (\text{floor}(\#DOWB\# * (10/7) / 10))))$ Week Day + 2 (BLUE) $((\#DOWB\# + 2) - (7 * (\text{floor}(\#DOWB\# * (10/6) / 10))))$
	Forecast Next Day Text		Standard Week Day (Sunday (1) to Saturday (7)) (WHITE) $\#WFACI\#$ Week Day + 1 (RED) $((\#WFACI\# + 1) - (7 * (\text{floor}(\#WFACI\# * (10/7) / 10))))$ Week Day + 2 (BLUE) $((\#WFACI\# + 2) - (7 * (\text{floor}(\#WFACI\# * (10/6) / 10))))$
	Back & Forth Motion [Pendulum]	Y Position:	$(219 + \sin((\#DWFSS\#) * 10))$ Or $(162 + \sin((\#DWFSS\#) * 70))$ Or $(160 + (120 * (\sin(\#DWE\# * 2 * \pi))))$ Or $\$(\text{round}(\text{floor}(2 * \#DWE\# + .5) / 2)) = (\text{floor}((2 * \#DWE\# + .5) / 2)) ? 160 + 120 * (-1 + 2 * ((2 * \#DWE\# + .5) - (\text{floor}(2 * \#DWE\# + .5)))) : 160 - 120 * (-1 + 2 * ((2 * \#DWE\# + .5) - (\text{floor}(2 * \#DWE\# + .5))))\$$
	Fade In/Out	Transparency	$((\sin(\#DWE\# * 5)) * 50 + 50)$ ccc
		Transparency	$\$Ds\# > 14 \& \& \#Ds\# < 27 ? 100 : 0\$$ Visible for specified time
	10sec blink	Transparency	$\$\text{floor}(\#DNOW\# / 10000) \% 2 = 0 ? 100 : 0\$$
	Font (Growing)	1sec	$((.2 + (\#DWE\#) \% 1) * 30)$.2 is a % of full font size of 30

Facer & WatchMaker WatchFaces

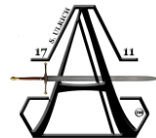
Quick Reference



Phone				
	Battery	Battery Progress	Fill Ratio	(clamp(#PBN#, 0, 100)/100)
			Transparency	\$#PBN#>0?100:0\$
	WiFi	Phone WiFi Level	Transparency	\$#PWL#=1?100:0\$ (1-5)
				#WSUNRISE24#
Other				
	Other Expressions for use and example	https://www.facer.io/watchface/nnnzoD8Ot5		((sin(#ZHR#*100))*50+50)
				(50+50*sin(2*pi*#Dsm#*#ZHR#/60))
				Tested and working fine! Thanks Mike, just replaced 2*pi for 6.28 (50+50*sin(6.28*#Dsm#*#ZHR#/60))
				Note: in watches without sensor the tag returns zero and break the formula, so is better to use a conditional like \$#ZHR#>0?(50+50*sin(6.28*#Dsm#*#ZHR#/60):(100-200*((#Dsm#/2)%0.5))\$ where if the tag is zero remains a steady half of a second pulse. Take out the spaces between the tags!!!!
			Rotation	(((((#WRH#+(#WRm#/60))/24)*(360))+225)+#DWFKS#)-(#DWFHS#+180))
			Rotation	((((((#WRH#+12)/2)+(#WRm#/60))/24)*(360))+225)+#DWFKS#)-(#DWFHS#+180))
			Rotation	(((((#WRH#+(#WRm#/60))/24)*(360))+225)+#DWFKS#)-(#DWFHS#+180))
			Rotation	(((((#WSH#+(#WSm#/60))/24)*(360))+135)+#DWFKS#)-(#DWFHS#+180))
			Rotation	((((((#WSH#+12)/2)+(#WSm#/60))/24)*(360))+135)+#DWFKS#)-(#DWFHS#+180))
			Rotation	(((((#WSH#+(#WSm#/60))/24)*(360))+135)+#DWFKS#)-(#DWFHS#+180))
			Rotation	((#DWFKS#)-(#DWFHS#+180))
			Rotation	((((((((((#DH#/24)+#DD#)*3)/1096)+#Dyyyy#)/(850481/10521600))-floor((((((((#Dsm#/60)+#Dm#)/60)+#DH#)/24)+#DD#)*3)/1096)+#Dyyyy#)/(850481/10521600)))*180)+11)+90)*(1))+85)
			Transparency	\$(#DH#+(#Dm#/60))>(#WRH#+(#WRm#/60))&&(#DH#+(#Dm#/60))<(#WSH#+(#WSm#/60))?0:100\$
		weather and daytime visible	Transparency	\$#DISDAYTIME#==true&&#WFACI#==01?100:0\$
				(round(#BLN#/10)*10)) Sets transparency from 100 to 0, step 10.
		Growing 60sec		\$floor(#DNOW#/500)%2=0?40:10\$

Facer & WatchMaker WatchFaces

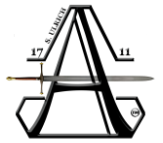
Quick Reference



				40=Max Font Size 10=Min Font Size
--	--	--	--	-----------------------------------

Facer & WatchMaker WatchFaces

Quick Reference



(Other Tuts)

Small obj orbiting edge of watch / second hand

X-position:

$(160 + \cos((\#DWFS\# - 90) * ((\pi)/180)) * 140)$

Y-position:

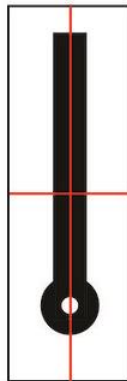
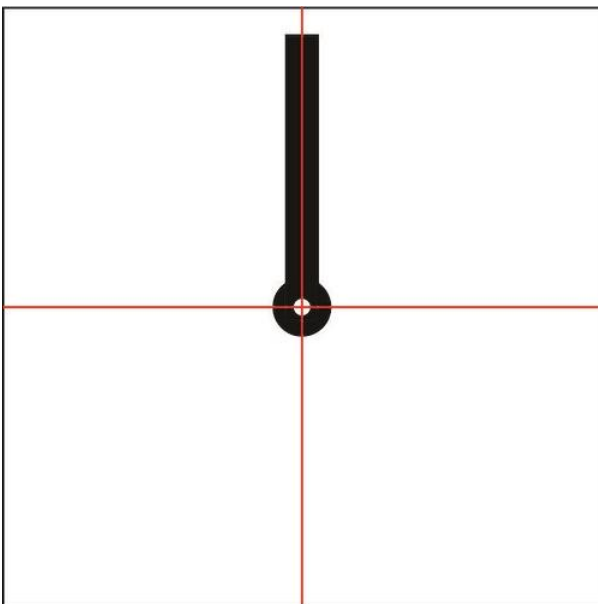
$(160 + \sin((\#DWFS\# - 90) * ((\pi)/180)) * 140)$

The 140 is what controls the spacing from the center

The DWFS tag is for seconds that "jump". if you want it to move smoothly use DWFSS.

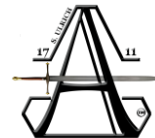
If you want to do one for hours use DWFK (I think). etc.

It is the way I use. when I make my own hand image I will crop the size of the image like the left one, so the hand will rotate at the center hole of the hand.



Facer & WatchMaker WatchFaces

Quick Reference



Basic Recipes

If you want to get started in more advanced techniques, we've put together a small collection of recipes for commonly used techniques.

Be sure to check out the [Community](#) for more techniques!

Width-Based Battery Indicator Bar

There are many ways to create a custom battery indicator, but one of the easiest is to use a shape & the width property to show the battery level for a watch or phone.

You can inspect this [watch face](#) to see it in action and look under the hood!

Recipe for Width-Based Battery Indication

Property: Width

```
((#BLN#)*1.55)
```

Rotate Any Element Over Time

While we provide essential elements for analog time, you might have an idea that makes a second/minute/hour hand out of a different element altogether (rotating text, shape, cat with laser eyes, etc). This recipe will show you how to rotate any element based on the time. For our example we use 'seconds': `#DWFSS#`, but you could also use a different [tag](#).

You can inspect this [watch face](#) to see it in action on a shape element!

Recipe for Rotating With Seconds

Property: Rotation

```
#DWFSS#
```

Back & Forth Motion

You can use the sine & cosine functions to make back & forth movement, as demonstrated by this laser cat's tongue:

You can inspect this [watch face](#) to see it in action and look under the hood!

Recipe for Back & Forth Motion

Property: Y Position

```
(219 + sin((#DWFSS#))*10)
```

Or

```
(162+sin((#DWFSS#))*70)
```

Advanced Animations

In order to help advanced users with a series of samples of possibilities, we've created the 'Expression Playground' watch face. This watch face isn't very useful for telling time, but you can use Inspection Mode to see how we achieve certain animations and effects with Facer.

How does this watch face work?

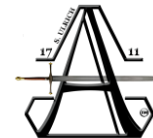
We'll go through a couple of specific expressions used in the watch face in the recipes below. You can open the watch in inspector mode to see how it all works:

 <https://www.facer.io/watchface/1WkGIWDQzf/inspect>

Feel free to make changes, nothing will get saved so you're free to play!

Facer & WatchMaker WatchFaces

Quick Reference



Fade In / Fade Out

These recipes cover 3 behaviors: Fade in when the watch wakes up, Fade out when the watch wakes up, or Fade In/Out on a loop.

Recipe for Fade In When the Watch Wakes:

Property: Transparency

$((\text{interpAccel}(\#DWE\#, 0, 2, 2)*100))$

Recipe for Fade Out When the Watch Wakes:

Property: Transparency

$(100-(\text{interpAccel}(\#DWE\#, 0, 2, 2)*100))$

Recipe for Looping Fade In & Fade Out:

Property: Transparency

$((\sin(\#DWE\#*5))*50+50)$

Pendulum Motion

Allow an element to move as if swinging from a string:

Recipe for Pendulum Motion:

Property: Rotation $(\sin(\#DWE\#*2)*50)$

“Breathing” Motion

Have an element cycle properties easing in & out, to create a ‘breathing’ effect

Recipe for Breathing Opacity:

Property: Transparency $((\sin(\#DWE\#*2)*60)+50)$

Recipe for Breathing Size:

Property: Radius $((\sin(\#DWE\#*2)*40)+50)$

Basic Orbit

Orbit an element in a circular motion

Recipe for Basic Orbit:

Property: X Position $(160 + \sin(\#DWFSS\#/4)*50)$

Property: Y Position $(160 + \cos(\#DWFSS\#/4)*50)$

Animated Battery Power Level Indicators

Here are the three basic possibilities we will speak about in this tutorial.

This tutorial should show you, in a simple way & at one place, how to elevate your watch faces with some basic expressions*.

Super Simple Percentage Number

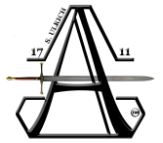
This is maybe something you are already using at your watch faces. Tag

#BLP#

shows your watch battery level as a text.

Facer & WatchMaker WatchFaces

Quick Reference



Color Changing

Changing of the colors according to the power percentage could be done with a duplication of element. I use usually 3 elements in three colors. Visibility of different colored elements could be changed at the transparency. Here an example for

Green

```
$#BLN#>40?100:0$
```

Yellow

```
$#BLN#>=20&&#BLN#<=40?100:0$
```

Red

```
$#BLN#<20?100:0$
```

In those expressions a tag `#BLN#` is using instead of `#BLP#` because we need a “pure” number to be able to compare it with another value. This color change could be apply for any battery level indicator in this tutorial (number, hand, or object)

Linear Battery Indicator

For this effect I use the standard “Rectangle” you could find in Facer.io Creator as Shape or I create an image especially for this purpose. In the case of an own picture, please be sure that your “indicator” **covers the full area of picture**, like this:

Red line shows only the end of the image. The second important thing is to set **correct align** according to the “growing” object in the right way:

So, the last thing we need is to make the “Height” of the object dynamic & appropriate to the battery level. If the “full battery level” means for example the number “48”, the final expression would be:

Height:

```
(#BLN#*0.48)
```

And it's done.

For changing the colors please use the same process as at the “Super Simple Percentage Number” part.

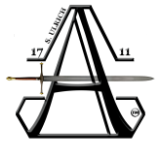
Hand Indicator

For those purposes I recommend to use of your own hands saved as PNG image (or another one with transparent background). Very important is the position of the hand according to the image. In the standard settings the pictures and objects rotate around the image centre. So, if you wish to have a nice and smooth movements, please check that the image centre is also the centre of your hand, like this:

Red line again shows only the end of the image. White background should be 100% transparent.
After you find the correct position of your hand according to the dial, e.g.:

Facer & WatchMaker WatchFaces

Quick Reference



The last thing is to set a dynamic rotation. If you are not sure what angle is your starting and the end position, please just rotate your hand into the limit positions like:

Now we have the range we are moving in. In this case 300 degree (of 360 degree for a full circle), starting (Tag `#BLN#` = 0) at -150 degree. The battery level is divided into 100 parts. It means that 1% change of battery level are 3 degrees change at the rotation. So the final expression will be:

Rotation:

```
(-150+(#BLN#*3))
```

Done!

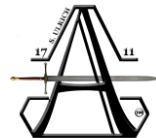
I hope, it could helps you and I would be happy, if you leave me a comment how well does it work at your watch face.

Here is an “inspection mode open” watch face, where you could see how does it work for linear battery and hand dials temperature:

*There are also another great tutorials and/ or explanations in the facer.io community done by e.g. [@Bertso](#) and [@jimmycheung](#) like [battery indicator](#) or [hand rotation](#). If you know about further tutorials, I will be happy to link it too.

Facer & WatchMaker WatchFaces

Quick Reference



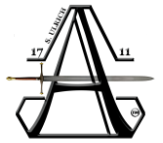
TAGS

Tags are dynamic values you can add to the dynamic attributes of a watch face layer. Facer tags are always surrounded by hashtags: #Ds#. Dynamic means the value can be different under different circumstances. For example, the #Ds# is the current second value, so every second this value changes.

TAGS	DEFINITION	EXAMPLE OUTPUT	
Date & Time			
#DWE#	Time elapsed since last watch face view (in seconds with 0.01 parts)	5.12	
#DNow#	Current timestamp	1550780363	
#DSync#	Timestamp at which watch face was synced	1550780132	
#Dy#	Year	2016	
#Dyy#	Short Year	16	
#Dyyyy#	Long Year	2016	
#DM#	Month in Year (Numeric)	11	
#DMM#	Month in Year (Numeric) with leading 0	11	
#DMMM#	Month in Year (Short String)	Nov	
#DMMMM#	Month in Year (String)	November	
#DW#	Week in Month	2	
#Dw#	Week in Year	45	
#DD#	Day in Year	313	
#Dd#	Day in Month	8	
#DdL#	Day in Month with leading 0	08	
#DIM#	Maximum days in the current month	30	
#DE#	Day of Week (Short String)	Tue	
#DES#	First letter of the day of Week	T	
#DOW#	Day of week int (0-6 where 0 is Sunday)	2	
#DOWB#	Day of week int (1-7 where 1 is Sunday)	3	
#DEEEE#	Day of Week	Tuesday	
#DF#	Day of Week in Month	2	
#Da#	AM/PM	PM	
#Db#	Hour in day using user's preferred format (12 hour or 24 hour clock). NOTICE: This is the recommended tag for any digital times.		
12			
#Dh#	Hour in Day (1-12). NOTICE: This should only be used in specific cases where #Db# does not work.	12	

Facer & WatchMaker WatchFaces

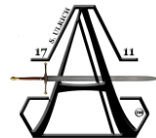
Quick Reference



#Dk#	Hour in Day (1-24). NOTICE: This should only be used in specific cases where #Db# does not work.	12	
#DH#	Hour in Day (0-23)	12	
#DK#	Hour in Day (0-11)	0	
#DHZ#	Hour in Day (leading zero) (00-23)	08	
#DkZA#	First Digit of the Hour in Day (0-2)	1	
#DkZB#	Second Digit of the Hour in Day (0-9)	2	
#DKZ#	Hour in Day (12 hour, leading zero) (00-11)	00	
#DhZ#	Hour in Day (12 hour, leading zero) (01-12)	09	
#DhZA#	First Digit (12 hour) of the in Day (0-1)	1	
#DhZB#	Second Digit (12 hour) of the in Day (0-9)	2	
#DhoT#	Value for Hour Rotation (12 hour) (30-360 with intervals of 30)	360	
#DhoTb#	Value for Hour Rotation (24 hour) (0-345 with intervals of 15)	180	
#DWFK#	Value for Hour Rotation (12 hour) (30-360 with intervals of 30)	360	
#DWFH#	Value for Hour Rotation (24 hour) (0-345 with intervals of 15)	180	
#DWFKS#	Smooth Value for Hour Rotation (12 hour) (30-390)	389	
#DWFHS#	Smooth Value for Hour Rotation (24 hour) (0-359.75 with 0.25 parts)	194.5	
#DhT#	String value for hour (12 hour)	twelve	
#DkT#	String value for hour (24 hour)	twelve	
#Dm#	Minute in Hour	58	
#DmZ#	Minute in Hour (leading zero)	08	
#DWFM#	Value for minute hand rotation (0-354 with intervals of 6)	348	
#DWFMS#	Smooth Value for minute hand rotation (0-359.9 with 0.1 parts)	351.8	
#DmT#	String value for minutes	fifty eight	

Facer & WatchMaker WatchFaces

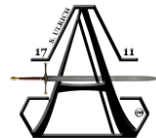
Quick Reference



	#DmMT#	String value for minutes (tens place)	fifty	
	#DmST#	String value for minutes (ones place)	eight	
	#Ds#	Second in minute	38	
	#DsZ#	Second in minute (leading zero)	09	
	#Dsm#	Second in minute plus milliseconds	38.805	
	#DseT#	Value for second hand rotation (0-354 with intervals of 6)	228	
	#DWFS#	Rotation value for second hand (0-354 with intervals of 6)	228	
	#DWFSS#	Smooth Rotation value for second hand (0-359.99399991780274)	232.829999946838	
	#DSMOOTH#	Smooth Rotation (boolean)	true	
	#Dz#	Timezone Abbreviation	PST	
	#Dzzzz#	Timezone Name (Long - if available)	PST	
	#DWR#	Day of Week Rotational Code (51.42857142857142(for sunday)-360(for saturday) with intervals of 51.42857142857142)	154.28571428571428	
	#DMR#	Day of Month Rotational Code (11.612903225806452(for 1st)-360(for 31th) with intervals of 11.612903225806452)	92.90322580645162	
	#DYR#	Day of Year Rotational Code (1-365(non-leap year),or 1-366(leap year))	313	
	#DMYR#	Month in Year Rotational Code (30-360 with intervals of 30)	330	
	#DUh#	Hour in Day (UTC) (1-12)	8	
	#DUk#	Hour in Day (UTC) (1-24)	20	
	#DUH#	Hour in Day (UTC) (0-23)	20	
	#DUK#	Hour in Day (UTC) (0-11)	8	
	#DUb#	Hours in day using user's preferred format (UTC) (12/24)		
	undefined			
	#DUHZ#	Hour in Day (leading zero) (UTC) (00-23)	05	

Facer & WatchMaker WatchFaces

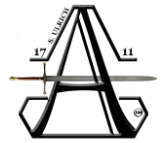
Quick Reference



	#DUkZ#	Hour in Day (leading zero) (UTC) (01-24)	02	
	#DUKZ#	Hour in Day (12 hour, leading zero) (UTC) (00-11)	08	
	#DUhZ#	Hour in Day (12 hour, leading zero) (UTC) (01-12)	08	
	#DUM#	Minute in Hour (UTC)	58	
	#DUMZ#	Minute in Hour (leading zero) (UTC)	09	
	#DUs#	Second in minute (UTC)	38	
	#DUsZ#	Second in minute (leading zero) (UTC)	08	
Device information				
	#BLP#	Watch Battery Level Percentage	46%	
	#BLN#	Watch Battery Level Integer	46	
	#BTC#	Watch Battery Temperature (°C)	31°C	
	#BTI#	Watch Battery Temperature (°F)	87°F	
	#BTCN#	Watch Battery Temperature (Celcius)	31	
	#BTIN#	Watch Battery Temperature (Fahrenheit)	87	
	#BS#	Watch Battery Charging Status	0	
	#ZLP#	Low Power Mode	null	
	#ZDEVICE#	Device Name	Smartwatch	
	#ZMANU#	Device Manufacturer	Facer	
	#ZISROUND#	Device Screen Shape (true, if circular)	false	
	#PBP#	Phone Level Percentage	79%	
	#PBN#	Phone Level Integer	79	
	#PWL#	Phone WiFi Level	3	
	#ZWC#	Number of watch face activations since synced	26	
Weather information				
	#WM#	Weather Units (F/M)	F	
	#WLC#	Weather Location	Los Angeles	
	#WTH#	Today's High	86	
	#WTL#	Todays' Low	63	
	#WCT#	Current Temp	84	
	#WCCI#	Current Condition Icon	03	
	#WCCT#	Current Condition Text	Fair	
	#WCHN#	Current Humidity Number	40	
	#WCHP#	Current Humidity Percentage	40%	
	#DISDAYTIME#	Returns true if time is after sunrise and before sunset	true	
	#WRh#	Sunrise hour (1-12)	5	

Facer & WatchMaker WatchFaces

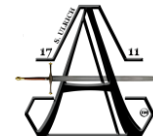
Quick Reference



#WRhZ#	Sunrise hour (leading zero) (01-12)	05	
#WRH#	Sunrise hour (0-23)	5	
#WRHZ#	Sunrise hour (leading zero) (00-23)	05	
#WRm#	Sunrise minute (0-59)	50	
#WRmZ#	Sunrise minute (leading zero) (00-59)	06	
#WSh#	Sunset hour (1-12)	8	
#WShZ#	Sunset hour (leading zero) (01-12)	08	
#WSH#	Sunset hour (0-23)	20	
#WSHZ#	Sunset hour (leading zero) (00-23)	01	
#WSm#	Sunset minute (0-59)	6	
#WSmZ#	Sunset minute (leading zero) (00-59)	06	
#WSUNRISE#	Time of sunrise	5:50 am	
#WSUNSET#	Time of sunset	8:06 pm	
#WSUNRISE24#	Time of sunrise (24)	5:50	
#WSUNSET24#	Time of sunset (24)	20:06	
#WSUNRISEH#	Hour of sunrise	5	
#WSUNRISEM#	Minute of sunrise	50	
#WSUNSETH#	Hour of sunset	8	
#WSUNSETM#	Minute of sunset	06	
#WSUNRISEH24#	Hour of sunrise (24)	5	
#WSUNSETH24#	Hour of sunset (24)	20	
#WFAH#	Forecast Day 1 High	86	
#WFAL#	Forecast Day 1 Low	63	
#WFACT#	Forecast Day 1 Condition Text	Clear	
#WFACI#	Forecast Day 1 Condition Icon	01	
#WFBH#	Forecast Day 2 High	85	
#WFB#	Forecast Day 2 Low	63	
#WFBCT#	Forecast Day 2 Condition Text	Sunny	
#WFBCI#	Forecast Day 2 Condition Icon	01	
#WFCH#	Forecast Day 3 High	85	
#WFCL#	Forecast Day 3 Low	63	
#WFCCT#	Forecast Day 3 Condition Text	Sunny	
#WFCCI#	Forecast Day 3 Condition Icon	01	
#WFDH#	Forecast Day 4 High	88	
#WFDL#	Forecast Day 4 Low	64	
#WFDCT#	Forecast Day 4 Condition Text	Sunny	

Facer & WatchMaker WatchFaces

Quick Reference



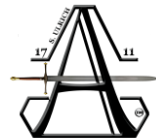
	#WFDCI#	Forecast Day 4 Condition Icon	01	
	#WFEH#	Forecast Day 5 High	86	
	#WFEL#	Forecast Day 5 Low	68	
	#WFECT#	Forecast Day 5 Condition Text	Mostly Sunny	
	#WFECI#	Forecast Day 5 Condition Icon	01	
Health/Fitness information				
	#ZSC#	Step Count	1556	
	#ZHR#	Average Heart Rate (bpm)	76	
Interactions				
	#SWISRUNNING#	Stopwatch Active State (boolean)	0	
	#SWEMS#	Stopwatch elapsed milliseconds	6034	
	#SWES#	Stopwatch elapsed seconds	6	
	#SWEM#	Stopwatch elapsed minutes	4	
	#SWEH#	Stopwatch elapsed hours	0.3	
Programmables				
Programmable tags can be updated using the programmable widgets of the Facer Creator. There are currently 6 variables, VAR_1 to VAR_6				
	#VAR_1#	Value for interactive variable 1	12	
	#VAR_1_T#	Timestamp at which interactive variable 1 was last updated	1550780123	

EXPRESSIONS

Math expressions can take your watch face to next level. Facer supports arithmetical operations as well as a handful of math functions. You can also drop facer tags into expressions to make them even more dynamic!

NOTE: Math expression require parentheses. So, this will work **(1+1)** and this will not **1+1** !

	OPERATOR	DEFINITION	EX EXPRESSION	EX OUTPUT	
Operators					
	+	Addition operator	(3+2)	5	
	-	Subtraction operator	(3-2)	1	
	*	Multiplication operator	(3*2)	6	
	/	Division operator	(3/2)	1.5	
	%	Modulo operator	(5%2)	1	
Constants					
	pi	Value of pi	(pi)	3.141592653589793	

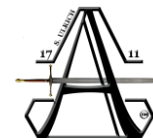


EXPRESSIONS

Math expressions can take your watch face to next level. Facer supports arithmetical operations as well as a handful of math functions. You can also drop facer tags into expressions to make them even more dynamic!

NOTE: Math expression require parentheses. So, this will work **(1+1)** and this will not **1+1** !

	OPERATOR	DEFINITION	EX EXPRESSION	EX OUTPUT	
	e	Value of e	(e)	2.718281828459045	
Functions					
	rand(min, max)	Random number generator	(rand(1, 10))	1	
	wakeRand(min, max)	Random number generator that's updated every time the watch face goes into active mode	(wakeRand(1, 10))	1	
	abs(number)	Absolute value function	(abs(-10))	10	
	sin(number)	Sine function	(sin(1))	0.8414709848078965	
	cos(number)	Cosine function	(cos(1))	0.5403023058681398	
	tan(number)	Tangent function	(tan(1))	1.5574077246549023	
	round(number)	Round function (get the closest integer)	(round(1.6))	2	
	ceil(number)	Ceiling function (gets the next highest integer)	(ceil(1.2))	2	
	floor(number)	Floor function - gets the next lowest integer)	(floor(1.6))	1	
	log(number)	Log function	(log(2))	0.6931471805599453	
	log2(number)	Log base 2 function	(log2(2))	1	
	log10(number)	Log base 10 function	(log10(2))	0.3010299956639812	
	sqrt(number)	Square root function	(sqrt(4))	2	
	cbrt(number)	Cube root function	(cbrt(8))	2	
	exp(number)	Gets E to the x power	(exp(2))	7.38905609893065	
	expm1(number)	Gets E to the x power minus 1	(expm1(2))	6.38905609893065	
	deg(radians)	Convert radians to degrees	(deg(1))	57.29577951308232	
	rad(degrees)	Converts degrees to radians	(rad(180))	3.141592653589793	

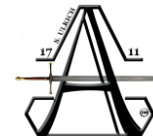


EXPRESSIONS

Math expressions can take your watch face to next level. Facer supports arithmetical operations as well as a handful of math functions. You can also drop facer tags into expressions to make them even more dynamic!

NOTE: Math expression require parentheses. So, this will work **(1+1)** and this will not **1+1** !

OPERATOR	DEFINITION	EX EXPRESSION	EX OUTPUT	
clamp(current, min, max)	Restrict 'current' value to the 'min' & 'max'	(clamp(#Ds#, 15, 45))	31	
squareWave(current, amplitude, period, xOffset)	Creates a square wave pattern out of 'current' value	(squareWave(#Ds#, 15, 10, 0))	15	
interpAccel(current, min, max, accelerationFactor)	Creates a ease-in transition with 'current' value	(interpAccel(#Ds#, 0, 60, 2))	0.2669444444444445	
interpDecel(current, min, max, accelerationFactor)	Creates a ease-out transition with 'current' value	(interpDecel(#Ds#, 0, 60, 2))	0.9454258487654321	
interpAccelDecel(current, min, max)	Creates a ease-in-out transition with 'current' value	(interpAccelDecel(#Ds#, 0, 60))	0.5261679781214715	
gyroX()	Outputs the X axis of the gyro sensor, accumulated into positional coordinates	3.9241372946780517	3.9241372946780517	
gyroY()	Outputs the Y axis of the gyro sensor, accumulated into positional coordinates	-1.0019503694835041	-1.0019503694835041	
accelerometerX()	Outputs the X axis of the accelerometer sensor, accumulated into positional coordinates	-5.273500685642811	-5.273500685642811	
accelerometerY()	Outputs the Y axis of the accelerometer sensor, accumulated into positional coordinates	7.629725700177308	7.629725700177308	



EXPRESSIONS

Math expressions can take your watch face to next level. Facer supports arithmetical operations as well as a handful of math functions. You can also drop facer tags into expressions to make them even more dynamic!

NOTE: Math expression require parentheses. So, this will work **(1+1)** and this will not **1+1** !

	OPERATOR	DEFINITION	EX EXPRESSION	EX OUTPUT	
	gyroRawX()	Outputs the raw X axis of the gyro sensor, which detects rotational acceleration	6.64500963775291	6.64500963775291	
	gyroRawY()	Outputs the raw Y axis of the gyro sensor, which detects rotational acceleration	-3.9204606214134596	-3.9204606214134596	
	accelerometerRawX()	Outputs the raw X axis of the accelerometer sensor	1.4595613230651239	1.4595613230651239	
	accelerometerRawY()	Outputs the raw Y axis of the accelerometer sensor	9.87964297713896	9.87964297713896	

CONDITIONALS

Conditionals are the most complicated and dynamic way to create a watch face.

They take this form:

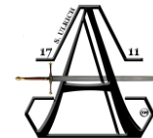
\$ EQUALITY TEST ? EXPRESSION IF TEST IS TRUE : EXPRESSION IF TEST IS FALSE \$

The first space is for our equality test, which is going to equal true or false. There are five types of operators available:

OPERATOR	DEFINITION	EX EXPRESSION	EX OUTPUT	
= or ==	Is equal to	(1=1) or (1==1)	true	
!=	Is not equal to	(1!=0)	true	
>	Is greater than	(2>1)	true	
<	Is less than	(0<1)	true	
>=	Is greater than or equal to	(2>=1)	true	
<=	Is less than or equal to	(0<=1)	true	

Facer & WatchMaker WatchFaces

Quick Reference



Now let's make our own. We'll fill it in step-by-step. Let's start with a blank slate. We'll develop a conditional to use for the Transparency attribute. All of the those dollar signs and question marks look confusing at first, but you'll get used to them (if you have a programming background, you probably already recognize the ternary operator!):

```
$ _____?_:_$
```

Now let's fill in that spot with a simple example. We're going to make an expression that will make a layer appear only before noon. This expression would be added to the opacity attribute of the layer. We'll use the #Da# and check if it's "AM":

```
$#Da#==AM?_:_$
```

Next, we'll fill in the EXPRESSION IF TRUE part of the conditional. When #Da# == "AM" is true, we want our layer to appear, so we'll want the opacity to be 100:

```
$#Da#==AM?100:_$
```

Lastly, we'll fill in the EXPRESSION IF FALSE part of the conditional. When #Da# == "AM" is false, we want our layer to disappear, so we'll want the opacity to be 0:

```
$#Da#==AM?100:0$
```

That's it! You just created your first conditional.

Now some advice:

Don't use " marks before and after texts used in conditions.

Don't use spaces (" ") before and after part of conditions (wrong: `$ #Da# == AM ? 100 : 0 $` right: `$#Da#==AM?100:0$`)

Boolean logic

For the ultra adventurous, the EQUALITY TEST portion of a Facer conditional also **supports a maximum of 3 basic Boolean operators in any given formula**. (Basic boolean operators are `||` and `&&`). Have fun!

A note about boolean operators in conditionals, it's important to avoid putting parentheses around equality and boolean expressions. Here are some examples:

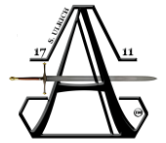
These will be ok:

```
$ 1 == 1 || 1 == 1 ? 1 : 0 $
```

```
$ (1+1) == (1) || (1) == (1) ? 1 : 0 $
```

Facer & WatchMaker WatchFaces

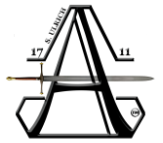
Quick Reference



These will have unexpected results:

```
$ (1 == 1) || (1 == 1) ? 1 : 0 $
```

```
$ (1 == 1 || 1 == 1) ? 1 : 0 $
```

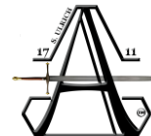


WatchMaker Tags

Tags are placeholders that can be used in expressions to make your watch face dynamic. They can be used in Lua scripts or simply be printed.

All properties that accept tags will open a tag picker dialog when clicking the little triangle next to the property value. The selected tag will be inserted at the cursor position

	Tag	Description	Example	
Date				
	{dd}	Day in month	9	
	{ddz}	Day in month (with leading 0)	09	
	{ddy}	Day in year	282	
	{ddw2}	Day of week	Th	
	{ddw}	Day of week	Thu	
	{ddww}	Day of week	Thursday	
	{ddw0}	Day of week (Sun = 0, Sat = 6)	5	
	{ddim}	Days in current month	31	
	{dn}	Month in year	1	
	{dnn}	Month in year (with leading 0)	01	
	{dnnn}	Month in year	Jan	
	{dnnnn}	Month in year	January	
	{dy}	Year (2 digits)	14	
	{dyy}	Year (4 digits)	2014	
	{dwm}	Week in month	4	
	{dw}	Week in year	40	
Time				
	{dh}	Hour in day (1-12)	9	
	{dh11}	Hour in day (0-11)	21	
	{dh24}	Hour in day (1-24)	9	
	{dh23}	Hour in day (0-23)	21	
	{dhutc12}	Hour in day UTC (12 hr)	8	
	{dhutc12z}	Hour in day UTC (12 hr with leading zero)	08	
	{dhutc24}	Hour in day UTC (24 hr)	20	
	{dhutc24z}	Hour in day UTC (24 hr with leading zero)	20	
	{dutcoff}	UTC Offset	+0100	
	{dht}	Hour in day text (1-12)	nine	
	{dh24t}	Hour in day text (1-24)	twenty one	

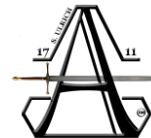


WatchMaker Tags

Tags are placeholders that can be used in expressions to make your watch face dynamic. They can be used in Lua scripts or simply be printed.

All properties that accept tags will open a tag picker dialog when clicking the little triangle next to the property value. The selected tag will be inserted at the cursor position

Tag	Description	Example	
{dhz}	Hour in day (1-12) (with leading zero)	09	
{dh11z}	Hour in day (0-11) (with leading zero)	21	
{dh24z}	Hour in day (1-24) (with leading zero)	09	
{dh23z}	Hour in day (0-23) (with leading zero)	21	
{dm}	Minute in hour	7	
{dmz}	Minute in hour (with leading zero)	07	
{dhtt}	Hour in day (1-12 tens)	0	
{dh11t}	Hour in day (0-11 tens)	0	
{dh11to}	Hour in day (0-11 ones)	9	
{dh24t}	Hour in day (1-24 tens)	0	
{dh24to}	Hour in day (1-24 ones)	9	
{dh23t}	Hour in day (0-23 tens)	0	
{dh23to}	Hour in day (0-23 ones)	9	
{dmz}	Minute in hour (with leading zero)	07	
{dmt}	Minute in hour (tens)	3	
{dmo}	Minute in hour (ones)	1	
{dmat}	Minute in hour text (all)	thirty one	
{dmtt}	Minute in hour text (tens)	thirty	
{dmot}	Minute in hour text (ones)	one	
{ds}	Second in minute	2	
{dsz}	Second in minute (with leading zero)	02	
{da}	AM/PM	AM	
{dss}	Milliseconds	32	
{dssz}	Milliseconds (with leading zeros)	032	
{dsps}	Seconds * 1000 + milliseconds	1207	
{depo}	Seconds since epoch	1495105593	
{dz}	Timezone	BST	

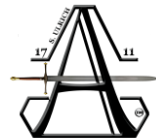


WatchMaker Tags

Tags are placeholders that can be used in expressions to make your watch face dynamic. They can be used in Lua scripts or simply be printed.

All properties that accept tags will open a tag picker dialog when clicking the little triangle next to the property value. The selected tag will be inserted at the cursor position

	Tag	Description	Example	
	{dtp}	Time (% 24 hours)	0.74	
	{drh}	Rotation value for hour (12h, adj for mins)	290	
	{drh24}	Rotation value for hour hand (24h, adj for mins)	145	
	{drh0}	Rotation value for hour hand (12h)	270	
	{drm}	Rotation value for min hand (adj for secs)	242	
	{drs}	Rotation value for second hand	156	
	{drss}	Rotation smooth value for second hand	156.2	
	{drms}	Rotation value for milliseconds	193.4	
Color Switcher				
	{ucolor}	Current Color	ff0000	
Counter				
	{c_elapsed}	Seconds elapsed since loaded	50	
	{c_0_100_2_st}	0 to 100 in 2s, then stop	50	
	{c_0_100_2_rp}	0 to 100 in 2s, then repeat	50	
	{c_0_100_2_rv}	0 to 100 in 2s, then reverse	50	
	{c_0_100_2_rv_2}	As above with 2s start delay	50	
Time Zone				
	{tz1l}	Extra Time Zone 1 Location	Los Angeles	
	{tz1o}	Extra Time Zone 1 Offset	- 8:00	
	{tz1t}	Extra Time Zone 1 Time	01:26	
	{tz1rh}	Extra Time Zone 1 Rotation hour hand	42	
	{tz1rh24}	Extra Time Zone 1 Rotation hour hand (24h)	24	
	{tz1rm}	Extra Time Zone 1 Rotation minute hand	160	
	{tz2l}	Extra Time Zone 2 Location	London	
	{tz2o}	Extra Time Zone 2 Offset	+ 0:00	
	{tz2t}	Extra Time Zone 2 Time	09:26	

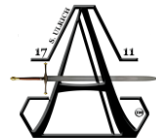


WatchMaker Tags

Tags are placeholders that can be used in expressions to make your watch face dynamic. They can be used in Lua scripts or simply be printed.

All properties that accept tags will open a tag picker dialog when clicking the little triangle next to the property value. The selected tag will be inserted at the cursor position

	Tag	Description	Example	
	{tz2rh}	Extra Time Zone 2 Rotation hour hand	283	
	{tz2rh24}	Extra Time Zone 2 Rotation hour hand (24h)	141	
	{tz2rm}	Extra Time Zone 2 Rotation minute hand	160	
	{tz3l}	Extra Time Zone 3 Location	Tokyo	
	{tz3o}	Extra Time Zone 3 Offset	+ 9:00	
	{tz3t}	Extra Time Zone 3 Time	18:26	
	{tz3rh}	Extra Time Zone 3 Rotation hour hand	190	
	{tz3rh24}	Extra Time Zone 3 Rotation hour hand (24h)	95	
	{tz3rm}	Extra Time Zone 3 Rotation minute hand	160	
	...up to 3 time zone locations can be specified in settings			
Battery				
	{bl}	Battery level	71	
	{blp}	Battery level %	71%	
	{br}	Rotation value for Battery level	250	
	{btc}	Battery temperature (C)	27	
	{btf}	Battery temperature (F)	81	
	{btcd}	Battery temperature (C) (percent)	27°C	
	{btfd}	Battery temperature (F) (percent)	81°F	
	{bc}	Battery charging status	Charging	
Phone				
	{pbl}	Battery Phone level	71	
	{pblp}	Battery Phone level %	71%	
	{pbr}	Rotation value for Battery Phone level	250	
	{pbtc}	Battery Phone temperature (C)	27	
	{pbtf}	Battery Phone temperature (F)	81	

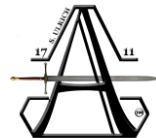


WatchMaker Tags

Tags are placeholders that can be used in expressions to make your watch face dynamic. They can be used in Lua scripts or simply be printed.

All properties that accept tags will open a tag picker dialog when clicking the little triangle next to the property value. The selected tag will be inserted at the cursor position

	Tag	Description	Example	
	{pbtcd}	Battery Phone temperature (C) (percent)	27°C	
	{pbtfd}	Battery Phone temperature (F) (percent)	81°F	
	{pbc}	Battery Phone charging status	Charging	
	{pws}	Phone Wifi Strength %	0.7	
	{pwc}	Phone Wifi Connected Y/N	Y	
Device				
	{aname}	Device name	GT1505	
	{aman}	Device manufacturer	samsung	
	{awname}	Watch name	Motorola 360	
	{around}	Is round?	true	
	{atyre}	Has flat tyre?	true	
	{abright}	Is bright?	true	
	{adimlo}	Dim mode supports lo-bit only?	false	
	{abss}	Milliseconds since bright (-1 for dim)	150	
	{alat}	Current latitude	40.7127	
	{alon}	Current longitude	74.0059	
	{alatd}	Current latitude (degrees)	40.7127°	
	{alond}	Current longitude (degrees)	74.0059°	
	{alatdd}	Current latitude (degrees + direction)	40.7127°N	
	{alonddd}	Current longitude (degrees + direction)	74.0059°E	
	{aalt}	Current altitude	104.800	
Stopwatch				
	{swh}	Stopwatch hours	0	
	{swm}	Stopwatch minutes	1	
	{sws}	Stopwatch seconds	25	
	{swss}	Stopwatch milliseconds (2 digits)	274	
	{swsss}	Stopwatch milliseconds (3 digits)	27	
	{swsst}	Stopwatch milliseconds total	1274	

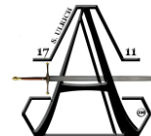


WatchMaker Tags

Tags are placeholders that can be used in expressions to make your watch face dynamic. They can be used in Lua scripts or simply be printed.

All properties that accept tags will open a tag picker dialog when clicking the little triangle next to the property value. The selected tag will be inserted at the cursor position

	Tag	Description	Example	
	{swr}	Stopwatch is running?	true	
	{swrm}	Stopwatch minute rotation	30	
	{swrs}	Stopwatch second rotation	45	
	{swrss}	Stopwatch millisecond rotation	47.2	
Weather				
	{wl}	Weather Location	London	
	{wt}	Current Temperature	13	
	{wth}	Today's High	14	
	{wtl}	Today's Low	12	
	{wtd}	Current Temperature (degrees)	13°C	
	{wthd}	Today's High (degrees)	14°C	
	{wtld}	Today's Low (degrees)	12°C	
	{wm}	Weather Units	c	
	{wct}	Current Condition Text	Rain	
	{wci}	Current Condition Icon	10d	
	{wh}	Current Humidity Number	87	
	{whp}	Current Humidity Percentage	87%	
	{wp}	Atmospheric Pressure	1003	
	{wws}	Wind Speed (mph)	6.2	
	{wwd}	Wind Direction (degrees)	140	
	{wwdb}	Wind Direction (NE)	NE	
	{wwdbb}	Wind Direction (NNE)	NNE	
	{wcl}	Cloudiness (%)	0.4	
	{wr}	Rain volume for last 3 hrs (mm)	30	
	{wsr}	Sunrise time	08:14	
	{wss}	Sunset time	17:15	
	{wsrp}	Sunrise time (% 24 hours)	0.3	
	{wssp}	Sunset time (% 24 hours)	0.7	
	{wmp}	Moon Phase (1=young, 5=full, 9=old)	3	
	{wml}	Weather manual location	New York	
	{wlu}	Weather last update	15:08:27	
	{wf0dt}	Forecast Day 0 Temp	11	
	{wf0dth}	Forecast Day 0 High	12	

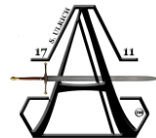


WatchMaker Tags

Tags are placeholders that can be used in expressions to make your watch face dynamic. They can be used in Lua scripts or simply be printed.

All properties that accept tags will open a tag picker dialog when clicking the little triangle next to the property value. The selected tag will be inserted at the cursor position

	Tag	Description	Example	
	{wf0dtl}	Forecast Day 0 Low	5	
	{wf0dct}	Forecast Day 0 Condition Text	Rain	
	{wf0dci}	Forecast Day 0 Condition Icon	10d	
	{wf1dt}	Forecast Day 1 Temp	11	
	{wf1dth}	Forecast Day 1 High	12	
	{wf1dtl}	Forecast Day 1 Low	5	
	{wf1dct}	Forecast Day 1 Condition Text	Rain	
	{wf1dci}	Forecast Day 1 Condition Icon	10d	
<i>Weather forecasts are available up to day 5. Fahrenheit can be set in settings.</i>				
Calendar				
	{cex}	Events Exist?	true	
	{c1t}	Event 1 Text	Gym	
	{c1ex}	Event 1 Exists?	true	
	{c1b}	Event 1 Begin	19:00	
	{c1br}	Event 1 Begin Rotation (12 hours)	210	
	{c1bp}	Event 1 Begin % 24 Hours	0.79	
	{c1e}	Event 1 End	20:00	
	{c1er}	Event 1 End Rotation (12 hours)	240	
	{c1ep}	Event 1 End % 24 Hours	0.83	
	{c1l}	Event 1 Location	Statue of Liberty	
	{c1c}	Event 1 Color	ff0000	
	{c1i}	Event 1 ID	520	
...calendars available for up to 10 unfinished events up to a week ahead (configurable in settings)				
Sensor				
	{ssc}	Step Count	1046	
	{shr}	Heart Rate	72	
	{sprs}	Barometric Pressure	0	
	{sax}	Accelerometer X	0.5	
	{say}	Accelerometer Y	0.5	

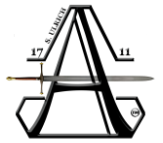


WatchMaker Tags

Tags are placeholders that can be used in expressions to make your watch face dynamic. They can be used in Lua scripts or simply be printed.

All properties that accept tags will open a tag picker dialog when clicking the little triangle next to the property value. The selected tag will be inserted at the cursor position

	Tag	Description	Example	
	{saz}	Accelerometer Z	0.5	
	{sgx}	Gyroscope X	0.5	
	{sgy}	Gyroscope Y	0.5	
	{sgz}	Gyroscope Z	0.5	
	{scr}	Compass for Rotation (needs negative)	-210	
	{sct}	Compass Display (0=N, 90=E, 180=S, 270=W)	210	
	{sctd}	Compass Display (degrees)	210°	
	{scb}	Compass Bearing (NE)	SW	
	{scbb}	Compass Bearing (NNE)	SSW	
	{sctdb}	Compass Display (degrees + NE bearing)	210° SW	
	{sctdbb}	Compass Display (degrees + NNE bearing)	210° SSW	
Complication (AWS 2.0+)				
	{m1text}	Complication 1 Text	e.g. News Text	
	{m1title}	Complication 1 Title	e.g. News Title	
	{m1value}	Complication 1 Value	e.g. 60, e.g. battery level	
	{m1min}	Complication 1 Min	e.g. 0, e.g. battery level	
	{m1max}	Complication 1 Max	e.g. 100, e.g. battery level	
	{m2text}	Complication 2 Text	e.g. News Text	
	{m2title}	Complication 2 Title	e.g. News Title	
	{m2value}	Complication 2 Value	e.g. 60, e.g. battery level	
	{m2min}	Complication 2 Min	e.g. 0, e.g. battery level	
	{m2max}	Complication 2 Max	e.g. 100, e.g. battery level	
	{m3text}	Complication 3 Text	e.g. News Text	
	{m3title}	Complication 3 Title	e.g. News Title	
	{m3value}	Complication 3 Value	e.g. 60, e.g. battery level	



WatchMaker Tags

Tags are placeholders that can be used in expressions to make your watch face dynamic. They can be used in Lua scripts or simply be printed.

All properties that accept tags will open a tag picker dialog when clicking the little triangle next to the property value. The selected tag will be inserted at the cursor position

	Tag	Description	Example	
	{m3min}	Complication 3 Min	e.g. 0, e.g. battery level	
	{m3max}	Complication 3 Max	e.g. 100, e.g. battery level	
Tasker				
	{t...}	Tasker Plugin Variable	Example Value	